

PTO/SB/21 (09-04)

Approved for use through 07/31/2006. OMB 0651-0031
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

09/2/23

**TRANSMITTAL
FORM**

(to be used for all correspondence after initial filing)

Total Number of Pages in This Submission

15

Application Number

09/827,739

Filing Date

April 6, 2001

First Named Inventor

John Sandham

Art Unit

2123

Examiner Name

Jason S. Proctor

Attorney Docket Number

45256.00004.CIP1

ENCLOSURES (Check all that apply)☐ Fee Transmittal Form☐ Fee Attached☐ Amendment/Reply☐ After Final☐ Affidavits/declaration(s)☐ Extension of Time Request☐ Express Abandonment Request☐ Information Disclosure Statement☒ Certified Copy of Priority Document(s)☐ Response to Missing Parts/
Incomplete Application☐ Response to Missing Parts
under 37 CFR 1.52 or 1.53☐ Drawing(s)☐ Licensing-related Papers☐ Petition☐ Petition to Convert to a
Provisional Application☐ Power of Attorney, Revocation
Change of Correspondence Address☐ Terminal Disclaimer☐ Request for Refund☐ CD, Number of CD(s) _____☐ Landscape Table on CD☐ After Allowance Communication to TC☐ Appeal Communication to Board
of Appeals and Interferences☐ Appeal Communication to TC
(Appeal Notice, Brief, Reply Brief)☐ Proprietary Information☐ Status Letter☒ Other Enclosure(s) (please Identify
below):

- Return Receipt Postcard

Remarks

Copy of Priority Document Application No. GB 9822074.2

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT

Firm Name

Paul, Hastings, Janofsky & Walker LLP

Signature

Printed name

Bradley D. Blanche

Date

July 6, 2005

Reg. No.

38,387

CERTIFICATE OF TRANSMISSION/MAILING

I hereby certify that this correspondence is being facsimile transmitted to the USPTO or deposited with the United States Postal Service with sufficient postage as first class mail in an envelope addressed to: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date shown below.

Signature

Typed or printed name

Angela Williams

Date

July 6, 2005

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

BEST AVAILABLE COPY

American LegalNet, Inc.
www.USCourtForms.com



INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

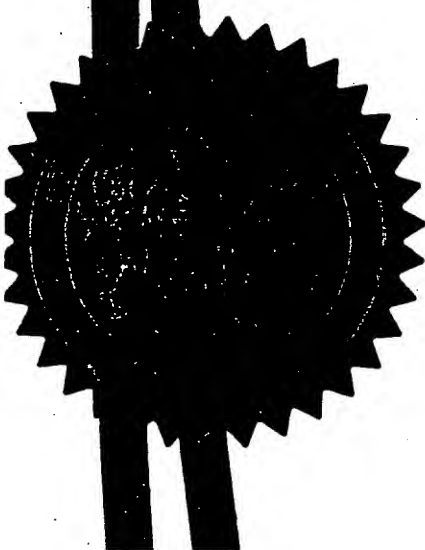
In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

M. C. Kie

Dated 12 April 2001

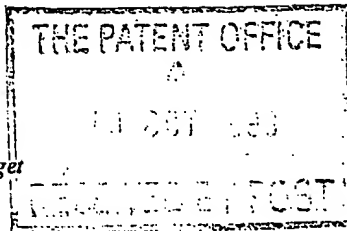


Patent Office

1200798 5396368-1 000354
P01/7700 0.00 - 9822074.2

Request for grant of a patent

(see the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)



The Patent Office

Cardiff Road
Newport
Gwent NP9 1RH

1.	Your reference	MH/M088261PGB		
2.	Patent application number (The Patent Office will fill in this part)	10 OCT 1998	9822074.2	
3.	Full name, address and postcode of the or of each applicant (underline all surnames)	THE VICTORIA UNIVERSITY OF MANCHESTER RUTHERFORD HOUSE MANCHESTER SCIENCE PARK PENCROFT WAY MANCHESTER M15 6SZ		
	Patents ADP number (if you know it)	6963516002		
	If the applicant is a corporate body, give the country/state of its incorporation	U.K. (UK)		
4.	Title of the invention	ENDIAN TRANSFORMATION		
5.	Name of your agent (if you have one)	Marks & Clerk		
	"Address for service" in the United Kingdom to which all correspondence should be sent (including the postcode)	Sussex House 83-85 Mosley Street Manchester M2 3LG		
	Patents ADP number (if you know it)	18004		
6.	If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (if you know it) the or each application number	Country	Priority application number (if you know it)	Date of filing (day/month/year)
7.	If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application	Number of earlier application	Date of filing (day/month/year)	
8.	Is a statement of Inventorship and of right to grant of a patent required in support of this request? (Answer 'Yes' if: a) any applicant named in part 3 is not an inventor, or b) there is an inventor who is not named as an applicant, or c) any named applicant is a corporate body. See note (d))	YES		

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form

Description

8

Claim(s)

2

Abstract

1

Drawing(s)

-

10. If you are also filing any of the following, state how many against each item.

Priority documents

Translations of priority documents

Statement of Inventorship and right to grant of a patent (Patents Form 7/77)

Request for preliminary examination and search (Patents Form 9/77)

1

Request for substantive examination (Patents Form 10/77)

Any other documents (Please specify)

11.

I/We request the grant of a patent on the basis of this application.

Signature Marks & Clerk

Date 9 October 1998

12. Name and daytime telephone number of person to contact in the United Kingdom

MATTHEW HOLMES
0161 236 2275

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

- a) If you need help to fill in this form or you have any questions, please contact the Patent Office on 0645 500505.
- b) Write your answers in capital letters using black ink or you may type them.
- c) If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.
- d) If you have answered 'Yes' Patents Form 7/77 will need to be filed.
- e) Once you have filled in the form you must remember to sign and date it.
- f) For details of the fee and ways to pay please contact the Patent Office.

Endian Transformation

This invention relates to an endian transformation method and system.

A problem commonly encountered by emulation systems, which run identical software on different computer processor chips is format incompatibility. One aspect of this incompatibility resides in the format in which strings of data (eg. 2-byte words or 4-byte words) are expressed. In many computer architectures, each byte of a 4-byte word has its own individual memory address; this gives rise to two possibilities for numbering the bytes within a word. In a big-endian convention, the word whose bytes are addressed (X , $X+1$, $X+2$ AND $X+3$) has its most significant byte addressed X , while in a little-endian convention, the address ordering is the reverse of this so that the least significant byte is addressed X and the most significant byte is addressed $X+3$. Other "endian formats" are known in which different conventions are observed for ordering the significance of bytes within words, but in most commercially available systems either the big-endian or little-endian convention is observed. The ordering of bits within each byte is the same whether the memory address convention is big-endian or little-endian.

Emulation systems are available which enable software (such as an operating system or an application program) of one endian format to operate on hardware of an opposite endian format. Generally, systems of this type convert each word between endian representations on a word-by-word basis. This conversion, when required frequently, introduces a significant overhead into the time required to perform a given task.

It is an object of the present invention to provide an efficient method and system to enable software of one endian format to run on hardware of a different endian format.

According to the invention there is provided a method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein

memory access addresses are transformed such that bytes stored in a memory addressed by a processor of the second type as a result of an instruction in which a byte order in accordance with the first convention is observed are distributed in a pattern which is a mirror image of the distribution pattern of the bytes which would result if the memory was addressed by a processor of the first type in response to the said instruction.

The invention also provides a method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, the order of the second convention being the reverse of the order of the first, wherein memory access addresses are transformed such that the offset between addresses of any two bytes stored in memory is unaltered by the transformation and the relative order of the addresses of any two bytes stored in the memory is reversed by the transformation.

The invention further provides a method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein memory access addresses are transformed such that strings of bytes in the first endian format which are stored successively by the processor operating in accordance with the second endian format aggregate in the same manner as the bytes would aggregate if the processor was of the first endian format and memory access addresses were not transformed.

The invention still further provides a method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein each memory access address B of string length L is transformed to the address $A-B-L+S$, wherein A is the total number of bytes allocated to a program, and S is the start address of the program.

Assuming a big-endian processor and a little-endian program, the address transformations ensure that bytes aggregate in the memory in a pattern which is a

mirror image of the pattern which would have resulted if the processor had been little-endian and no address transformation had been performed. The invention will operate in the same manner for a little-endian processor and a big-endian program. It is important to note that the transformation has no effect on the ordering of bits within each byte. The result is a system which provides a considerable time saving when compared to known endian conversion methods, which convert each string of bytes between endian representations each time that string is used.

According to the invention there is provided an endian transformation system, the system comprising means for transforming an address location of a code represented in a first endian format into an address in a second endian format, the transformation comprising introducing an offset into the address, the size of the offset being determined from the difference between the address location of the code and a predefined address location.

A specific embodiment of the invention will now be described by way of example only.

The following two assembly code store instructions

```
movl $0xaabbccdd,[0]
```

```
movl $0x11223344,[2]
```

will have the following effect in a little-endian architecture:

1st Store		2nd Store	
	23		23
	—		—
	6		6
	5	11	5
	4	22	4
aa	3	33	3
bb	2	44	2
cc	1	cc	1
dd	0	dd	0

The first store instruction stores the least significant byte (dd) of the first 4-byte word (aabbccdd) at address '0', the second least significant byte at address '1', etc. The second store instruction stores the least significant byte (44) of the second 4-byte word (11223344) at address '2', the second least significant byte at address '3', etc. Thus, the effect of storing the second 4-byte word is to overwrite the prior contents of addresses '2' and '3', and the two most significant bytes of the first 4-byte word 'aa' and 'bb' are lost.

If the same two store instructions are used in a big-endian architecture, the most significant byte (aa) of the first 4-byte word will be stored at address '0', the next most significant byte (bb) at address '1', etc. The second store instruction will overwrite the prior contents of addresses '2' and '3', as before, and the two least significant bytes of the first 4-byte word 'dd' and 'cc' will be lost. Thus, when a big-endian architecture is used, the contents of an addressed memory resulting from assembly code store instructions will differ from the contents of an equivalent memory when a little-endian architecture is used.

The invention allows both big-endian and little-endian words to be stored in such a way that any given store instruction will result in the same bytes being stored in both architectures, although the order of the bytes is reserved. This is achieved by transforming memory access addresses such that the pattern of bytes stored in a

memory addressed by for example a big-endian processor is a mirror image of the pattern which would have resulted if the memory had been addressed without transformation by a little-endian processor.

In the case of the assembly code store instructions given above, in order to accommodate a big-endian architecture, the two 4-byte words are stored in accordance with the present invention at the uppermost available addresses as shown below:

1st Store		2nd Store	
dd	23	dd	23
cc	22	cc	22
bb	21	44	21
aa	20	33	20
	19	22	19
	18	11	18
	-----		-----
	0		0

To preserve information in the memory, the second of the two stores places the second 4-byte word at a lower address than the first 4-byte word, thus overwriting the same 2-byte word (0xaabb) of information as in the little-endian architecture. The effect in terms of the bytes stored in the memory of the assembly code instructions in the little-endian architecture is thus duplicated in the big-endian architecture, although the order of the bytes is reversed.

The above exemplification of the system in accordance with the present invention in relation to an unaligned store instruction demonstrates the flexibility of the system. The system may also be used for aligned store instructions.

The address transformations used as described above to preserve information in the big-endian architecture are:

access type	adjustment
word (4-byte)	$\text{addr}' = [20 - \text{addr}]$
word (2-byte)	$\text{addr}' = [22 - \text{addr}]$
byte	$\text{addr}' = [23 - \text{addr}]$

This generalises to:

access type	general adjustment	where
word (4-byte)	$\text{addr}' = \text{endianAdj_L} - \text{addr}$	$\text{endianAdj_L} = \text{progSize} - 4$
word (2-byte)	$\text{addr}' = \text{endianAdj_W} - \text{addr}$	$\text{endianAdj_W} = \text{progSize} - 2$
byte	$\text{addr}' = \text{endianAdj_B} - \text{addr}$	$\text{endianAdj_B} = \text{progSize} - 1$

Thus, using the generalisation shown in the above table, the following operations in a little-endian architecture:

```
movw $0xaabb,[1]
```

```
movl [1],%eax
```

will have the same effect as the following operations in a big-endian architecture:

```
movw $0xaabb,[22-1]
```

```
movl [20-1],%eax
```

The effect of the above commands is shown below:

little-endian		big-endian	
3rd Store		3rd Store	
	23	dd	23
	22	bb	22
11	5	aa	21
22	4	33	20
33	3	22	19
aa	2	11	18
bb	1		
dd	0		0

The invention introduces one extra arithmetic operation for every load/store instruction. However, many instructions which access memory use address expressions which contain constant offsets such as:

`addl %edx,0x8(ebp,eax,4)`

which represents the effective address:

`ebp+eax*4+8.`

This expression, after memory access transformation in accordance with the invention has been applied, becomes:

`endianAdj_L-(ebp+eax*4+8).`

Folding the constants of the expression can be used to give:

`(endianAdj_L-8)-(ebp+eax*4).`

Thus, folding allows those terms which may be calculated at translation time to separated from those terms which are held in registers and are unknown at translation time. Since the term 'endianAdj_L' is known at translation time its effect is calculated before run time, and the memory access transformation will not cause a loss of performance at run time. Thus, in general the big-endian transformation of the invention incurs no extra overhead for the majority of memory accesses.

A subject machine program (or operating system) is treated as if it is loaded contiguously from address 0, while internally being stored as a mirror image, as shown below:

Actual Memory Configuration		Memory Configuration intended by the assembly code	
(big-endian)		(little-endian)	
c7	23		23
45	22	
f8	21	00	6
03	20	00	5
00	19	00	4
00	18	03	3
00	17	f8	2
	45	1
	0	c7	0

If the assembly code specifies access to the 4-byte value 0x00000003 at memory location 3, using the same program size as the previous examples, this memory access becomes $\text{endianadj_L-3} = (\text{progSize-4})-3 = (24-4)-3 = 17$, which is the address in the big-endian mirror image of the value required.

Whereas the above examples illustrate use of the invention in transforming code intended for a little-endian architecture so that it will run on a big-endian system, the invention could be used to transform big-endian code to run on a little-endian system. Indeed, the invention can be used to transform between any two endian systems which are byte reversals of one another.

The endian transformation method may be used as part of a complete emulation system.

Claims

1. A method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein memory access addresses are transformed such that bytes stored in a memory addressed by a processor of the second type as a result of an instruction in which a byte order in accordance with the first convention is observed are distributed in a pattern which is a mirror image of the distribution pattern of the bytes which would result if the memory was addressed by a processor of the first type in response to the said instruction.
2. A method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, the order of the second convention being the reverse of the order of the first, wherein memory access addresses are transformed such that the offset between addresses of any two bytes stored in memory is unaltered by the transformation and the relative order of the addresses of any two bytes stored in the memory is reversed by the transformation.
3. A method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein memory access addresses are transformed such that strings of bytes in the first endian format which are stored successively by the processor operating in accordance with the second endian format aggregate in the same manner as the bytes would aggregate if the processor was of the first endian format and memory access addresses were not transformed.

4. A method for emulating a processor of a first type which observes a first convention for ordering the significance of bytes within words on a second type of processor which observes a second convention for ordering the significance of bytes within words, wherein each memory access address B of string length L is transformed to the address $A-B-L+S$, wherein A is the total number of bytes allocated to a program, and S is the start address of the program.
5. An endian transformation system, the system comprising means for transforming a memory access address in accordance with a method according to any preceding claim.
6. A method of endian transformation substantially as hereinbefore described.
7. A system for providing endian transformation substantially as hereinbefore described.

Abstract

A method for emulating a processor of a first endian type on a processor of a second endian type, wherein each memory access address B of string length L is transformed to the address $A-B-L+S$, wherein A is the total number of bytes allocated to a program, and S is the start address of the program.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☒ **BLACK BORDERS**
- ☐ **IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- ☐ **FADED TEXT OR DRAWING**
- ☐ **BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- ☐ **SKEWED/SLANTED IMAGES**
- ☐ **COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- ☐ **GRAY SCALE DOCUMENTS**
- ☐ **LINES OR MARKS ON ORIGINAL DOCUMENT**
- ☐ **REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- ☐ **OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.